Robust Grid Computing using Peer-to-Peer Services

A. Sussman, P. Keleher, B. Bhattacharjee, D. Richardson, D. Wellnitz

Personnel

- Alan Sussman - PI - Grid computing
- Pete Keleher - co-PI - P2P algorithms
- Bobby Bhattacharjee - co-PI - P2P algorithms
- Derek Richardson - co-PI - astrononomy applications
- Dennis Wellnitz - co-PI - Deep Impact applications
- Michael Marsh - research scientist - implementation and supervising student implementation work
- Jik-Soo Kim - graduate student - matchmaking algorithms
- Beomseok Nam - graduate student - basic P2P algorithms and implementation
- San Ratanasanya - visiting graduate student (from Thailand) - client implementation

Activities and Findings

The initial work on the project has been the investigation of effective strategies for matching job requests to available resources using peer-to-peer (P2P) techniques in a distributed desktop Grid environment. We have extensively simulated and analyzed the behavior of two very different strategies for both satisfying job requirements (specified as minimum resource needs, such as memory, disk space, etc.) and balancing load across available resources. The Rendezvous Node Tree (RNT) approach uses the randomization characteristics of a Chord-based Distributed Hash Table (DHT) to spread jobs across all available resources, and then institutes a search from a randomly selected node to find a nearby node (in the Chord space) that meets the job requirements. The Content Addressable Network (CAN) based approach models the matchmaking problem as a multidimensional space in the various resources. Each CAN dimension corresponds to the values of one resource, and a node in the Grid is then mapped into the CAN based on its resource capabilities. Jobs are then routed using the CAN mechanisms to the proper part of the CAN based on their resource requirements, and then a local search is initiated to balance load in the part of the CAN space the job maps to. For both approaches, the goal is to both meet a job's requirements and balance load effectively across the available nodes in the desktop Grid. All the algorithms are completely decentralized, with only limited amounts of information passing among the nodes in the Grid, and nodes can arrive and leave (voluntarily or by failing) at any time, with the overall system dealing with such events as in other peer-to-peer dynamic systems.

The work has been presented at the 2006 International Grid Computing Conference, where it received a best paper award. The presentation was given by graduate student Jik-Soo Kim, and was very well received. A journal version of the paper has been submitted to a special issue of the International Journal of Grid Computing: Theory, Methods & Applications devoted to the Grid Computing conference. A graduate student on the project, Jik-Soo Kim, received a second place award at the ACM Student Research Competition at the SC'06 conference, for his poster and presentation on his work on the project. Another paper on improved matchmaking algorithms was recently accepted for publication in the High Performance Distributed Computing conference (HPDC 2007) in June 2007, and a project overview paper will appear at a workshop at the International Parallel and Distributed Processing Symposium (IPDPS) in March 2007.

The major finding from the study of the two P2P matchmaking algorithms is that the CAN-based algorithms perform better in most cases, both for load balancing and for matchmaking costs. Those results were reported in the Grid Computing Conference paper. The one failing of the CAN-based algorithm comes from workloads that are heavily clustered, meaning that many jobs are submitted in a short time period with similar resource requirements (e.g., for a parameter sweep set of physical simulations). That is because all of those jobs get mapped to the same part of the CAN space. That has led us to investigate more effective load balancing strategies for the CAN-based matchmaking algorithms in ongoing work. The overall result of the study of matchmaking algorithms is that we have chosen to use the CAN-based algorithm, with enhancements for robustness and better load balancing, in our peer implementation, which is still in progress.

Implementation status

We currently have working implementations of the basic peer-to-peer protocols for CAN (content-addressable networks), using the algorithms described in the papers listed below. The basic matchmaking algorithm has also been implemented on top of the basic P2P protocol, and implementation of the extended matchmaking algorithm that should provide better load balancing characteristics is under way.

We also have implemented a simple client program for submitting jobs to a peer, and have been using that for system testing. A more functional client, with a GUI and enhanced job monitoring capabilities, is currently being implemented to make it easier to submit jobs. A simple but effective authentication server has also been implemented to enable a set of users to share resources within the group, both to add machines and to authenticate users who want to submit jobs.

Important implementation work that still needs to be performed includes protection mechanisms to limit access to local resources for jobs running on a peer, and for more effectively dealing with job input and output (currently input data is copied from the client machine to the machine that runs the job and output data is copied back, using an ftp-like protocol).

Training and Development

The project has provided training for the graduate students on the project, getting them working with desktop grid and peer-to-peer technology. The project should lead to at least one PhD dissertation on work done for the project. The project has also brought together faculty in Computer Science and Astronomy with different skills and expertise. The computer scientists have and will learn more about the astronomy applications and their requirements as they are providing the initial workload characteristics for the desktop grid system, and the astronomers are learning what the computer science tools can do to help them share resources with their colleagues within Maryland and at other institutions. The collaboration among the computer science faculty has been very fruitful, sharing expertise about peer-to-peer techniques and Grid computing that no one person has.

The project has also inspired the PI to offer a graduate class on the intersection of Grid and Peer-to-Peer computing in Spring 2007, which is training both computer science graduate students and computational science graduate students in the Maryland Applied Math program in the technologies and potential applications. The cross fertilization between computer science and computational science students is proving fruitful for both sides, especially since they are doing significant programming projects together.

Publications

Journal

1. J.-S. Kim, B. Nam, P. Keleher, M. Marsh, B. Bhattacharjee and A. Sussman. “Trade-offs in Matching Jobs and Balancing Load for Distributed Desktop Grids”, submitted to *International Journal of Grid Computing: Theory,*

Methods & Applications, 2007.

Conference

1. J.-S. Kim, P. Keleher, M. Marsh, B. Bhattacharjee and A. Sussman. "Using Content-Addressable Networks for Load Balancing in Desktop Grids", to appear in *Proceedings of the 16th IEEE International Symposium on High Performance Distributed Computing (HPDC-16)*, June 2007

2. J.-S. Kim, B. Nam, M. Marsh, P. Keleher, B. Bhattacharjee, D.C. Richardson, D. Wellnitz and A. Sussman. "Creating a Robust Desktop Grid using Peer-to-Peer Services", to appear in *Proceedings of the 2007 NSF Next Generation Software Workshop*, March 2007.

3. J.-S Kim, B. Nam, P. Keleher, M. Marsh, B. Bhattacharjee and A. Sussman. "Resource Discovery Techniques in Distributed Desktop Grid Environments", *Proceedings of the 7th IEEE/ACM International Conference on Grid Computing - GRID 2006*, September 2006. Best paper award.

Future plans

We will continue to refine and develop the algorithms for matching resource requests to available resources. Those algorithms are being implemented in the peer software, and will be thoroughly tested and evaluated in the upcoming year. Once we are confident that the matchmaking algorithms are robust, and that the rest of the peer software implementation is reasonably stable, we will deploy the software within the project, both in computer science and to the astronomy collaborators. The goal is to spend a lot of effort in the upcoming year on making the peer software reliable, so that it will be useful to the astronomy collaborators and allow for monitoring and measurement of the behavior of the system under real workloads. When we are confident of the reliability and performance of the peer software, we will then investigate deploying outside the project, to current collaborator of both the astronomy co-Is and to other computational science collaborators of the PI not directly involved in the project.

The other major effort will be to characterize the different types of workloads that the system will be exposed to, to be able to simulate the behavior of the system under such workloads, for much larger resource configurations than are likely to become available to the project in the near future. We plan to mine Condor logs both within clusters currently running in the Maryland Institute for Advanced Computer Studies (UMIACS) that serve a diverse community of computational scientists, and on a cluster run by Astronomy co-I Richardson that is used for many types of astronomy simulations. We will also perform more qualitative studies of application characteristics (e.g., through interviews with scientists), to gain a better understanding of the workloads that the P2P resource sharing system we are building will be exposed to. In turn, all of the workload information will be used to improve the overall performance of the system under those workloads.